(19)日本国特許庁(JP)　(12)公開特許公報(A)　(11)特許出願公開番号
特開2004-53969
(P2004-53969A)
(43)公開日　平成16年2月19日(2004.2.19)

(51)Int.Cl.[7]		F I			テーマコード（参考）
G09C	1/04	G09C	1/04		5J104
H04L	9/34	H04L	9/00	681	

審査請求 有　請求項の数 11　OL　（全 13 頁）

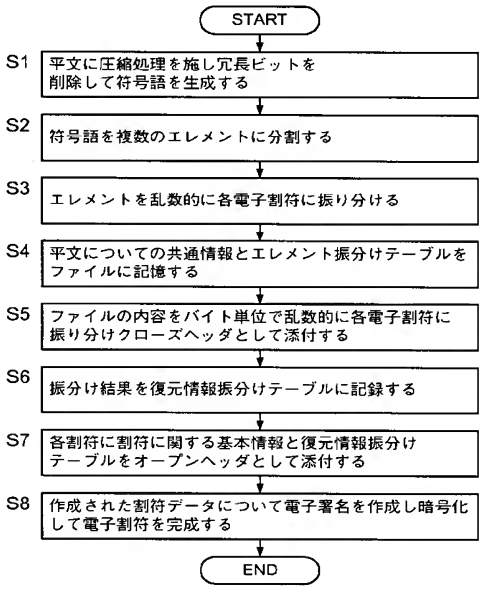
(21)出願番号　特願2002-211931(P2002-211931)
(22)出願日　平成14年7月22日(2002.7.22)

(71)出願人　500401453
グローバルフレンドシップ株式会社
東京都新宿区四谷四丁目１３番地
(74)代理人　100104341
弁理士　関　正治
(72)発明者　保倉　豊
東京都渋谷区幡ヶ谷１－１１－１３－５０６

Fターム(参考)　5J104 AA12

(54)【発明の名称】電子割符生成方法およびプログラム

(57)【要約】
【課題】極めて高度な機密性を有し信頼性の高い電子割符を簡単に生成する電子割符生成方法を提供する。
【解決手段】平文を読出して圧縮符号化し冗長なビットパターンを消した符号語を生成して、K個のエレメントに切り刻み、各エレメントを乱数に基づいてM個の割符ファイルに振り分けて格納し、その振り分け方法をエレメント割振りテーブルに記録し、エレメント割振りテーブルをM個に分割し割符ファイルにクローズヘッダとして追加し、割符ファイルごとのクローズヘッダに分配された エレメント割振りテーブルの分割片の配置リストをオープンヘッダとして割符ファイルに追加することにより、割符ファイルに電子割符を生成し必要に応じて出力する。
【選択図】　図１



【特許請求の範囲】
【請求項１】
符号語ファイルと予め決めたＫ個のエレメントファイルと予め決めたＭ個の割符ファイルと平文を格納した平文ファイルとエレメント割振りテーブルをコンピュータメモリ上に準備し、前記平文ファイルから平文を読み出して圧縮符号化し冗長なビットパターンを消した符号語を生成して前記符号語ファイルに格納し、該符号語を読み出してＫ個のエレメントに分割しエレメントごとにＫ個の前記エレメントファイルに格納し、各エレメントを読み出しては乱数に基づいてＭ個の前記割符ファイルのいずれかに振り分けて格納し、その振り分け方法を前記エレメント割振りテーブルに記録し、該エレメント割振りテーブルを読み出してＭ個に分割し前記割符ファイルにクローズヘッダとして追加し、該割符ファイルごとのクローズヘッダに分配された前記エレメント割振りテーブルの分割片の配置リストをオープンヘッダとして前記割符ファイルに追加することにより、前記割符ファイルに電子割符を生成することを特徴とする電子割符生成方法。
【請求項２】
前記エレメント割振りテーブルをＭ個に分割するときは、予め決めた大きさのビット長を単位として分割し、乱数に基づいてＭ個の割符に分配することを特徴とする請求項１記載の電子割符生成方法。
【請求項３】
前記平文から前記符号語を作成するときにおいて、前記圧縮符号化する前または後で適当な長さの乱数データを付加することを特徴とする請求項１または２記載の電子割符生成方法。
【請求項４】
前記平文の先頭部分に適当な長さの乱数データを付加した後に前記圧縮符号化して前記符号語を作成することを特徴とする請求項１または２記載の電子割符生成方法。
【請求項５】
前記符号語の先頭部分を適当な長さの乱数データでマスクして前記符号語を変成し、該乱数データを前記クローズヘッダに記録することを特徴とする請求項１から４のいずれかに記載の電子割符生成方法。
【請求項６】
前記エレメントは１ビット長であることを特徴とする請求項１から５のいずれかに記載の電子割符生成方法。
【請求項７】
前記エレメントをＭ個の前記割符ファイルに割り振るときに使用する乱数はＮビットごとに同じ配列を繰返した疑似乱数であることを特徴とする請求項１から６のいずれかに記載の電子割符生成方法。
【請求項８】
前記エレメントをＭ個の前記割符ファイルに割り振るときに使用する乱数は自然界に発生する電気的な雑音に基づくものであることを特徴とする請求項１から６のいずれかに記載の電子割符生成方法。
【請求項９】
さらに、配布先ごとに分配ファイルを準備して、前記Ｍ個の割符ファイルのそれぞれを異なる２個の分配ファイルに重複して含ませるように組合わせて、３以上の配布先に分配するようにしたことを特徴とする請求項１記載の電子割符生成方法。
【請求項１０】
符号語ファイルと予め決めたＫ個のエレメントファイルと予め決めたＭ個の割符ファイルと平文を格納した平文ファイルとエレメント割振りテーブルをコンピュータメモリ上に準備する手順と、前記平文ファイルから平文を読み出して圧縮符号化し冗長なビットパターンを消した符号語を生成して前記符号語ファイルに格納する手順と、該符号語を読み出してＫ個のエレメントに切り刻みエレメントごとにＫ個の前記エレメントファイルに格納する手順と、各エレメントを読み出しては乱数に基づいてＭ個の前記割符ファイルのいずれ

かに振り分けて格納する手順と、その振り分け方法を前記エレメント割振りテーブルに記録する手順と、該エレメント割振りテーブルを読み出してＭ個に分割し前記割符ファイルにクローズヘッダとして追加する手順と、該割符ファイルごとのクローズヘッダに分配された前記エレメント割振りテーブルの分割片の配置リストをオープンヘッダとして前記割符ファイルに追加する手順をコンピュータに実行させて、前記割符ファイルに電子割符を生成するようにしたプログラム。

【請求項１１】

請求項１０記載のプログラムを記録したコンピュータ読み取り可能な記憶媒体。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】

本発明は、電子情報を安全に送信したり、当事者間で取引などの確認を客観的に行うために使用する電子割符の生成方法およびコンピュータプログラムに関する。

【０００２】

【従来の技術】

ネットワークを通じて電子情報を交換すると、内容の改　、書き換え、すり替えが可能で、しかもその痕跡が残らず事後に検知することが難しいという難点があった。また、通信路の途中で盗取して利用することも比較的容易であった。

従来、このような場合に電子情報を暗号化して送付することにより安全を確保する方法が用いられてきた。しかし、暗号化方法は、送付したい電子情報の全てが通信データに含まれているため、通信が漏洩した時には極めて能力の高い侵害者ならば電子情報の解読や改　も可能である。

【０００３】

また、通信路を介して契約を交す例が多くなってきているが、電子ファイル化された契約内容は容易に改変することができ、知らない間にあるいは故意に改　されたりすると、契約の実行をめぐって争いが生じるおそれがある。このため、契約内容を随時確認できる簡便なシステムが要請されている。

このような通信における漏洩を防ぎまた契約内容などを客観的に確認できる手法として、本出願人が既に特開２０００－５９６５４３に開示した電子割符法がある。

【０００４】

ここで、電子割符とは電子情報を２以上に分割した片割れをいい、全ての電子割符を集合して統合しなければ元の電子情報を復元できないようになっている。開示された電子割符法は、図１２に概念を示すように、元データを多数のエレメントに分割し、エレメントを乱数に基づいて組合わせ、同じ集団に属するエレメントを乱数に基づいて並べ直すことによりいくつかの電子割符を作成し、電子割符をそれぞれ別のルートで送付したり保管して、必要なときに電子割符を集め、逆の手順でエレメントを並べ直すことにより元の情報を復元して使用するものである。

【０００５】

一部の電子割符を窃取しても元の情報を復元することができず安全である。

また、当事者同士が電子割符を分ち持つことによって、一方の保管した部分が改　されても、当事者の電子割符を集合させて統合することにより意味のある情報が復元できるかを見ることにより、改　の事実を検知することができる。

したがって、契約内容を電子割符化して分割保管しておくことにより、取引の安全が確保できる。

【０００６】

【発明が解決しようとする課題】

そこで、本発明が解決しようとする課題は、極めて高度な機密性を有し信頼性の高い電子割符を簡単に生成する電子割符生成方法およびコンピュータプログラムを提供することである。

【０００７】

【課題を解決するための手段】
上記課題を解決するため、本発明の電子割符生成方法は、コンピュータメモリ上の平文ファイルから平文を読み出して圧縮符号化し冗長なビットパターンを消した符号語を生成して符号語ファイルに格納し、この符号語を読み出してK個のエレメントに分割しエレメントごとに予め決めたK個のエレメントファイルに格納し、各エレメントを読み出しては乱数に基づいてM個の割符ファイルのいずれかに振り分けて格納し、その振り分け方法をエレメント割振りテーブルに記録し、このエレメント割振りテーブルを読み出して予め決めたM個に分割し割符ファイルにクローズヘッダとして追加し、割符ファイルごとのクローズヘッダに分配された エレメント割振りテーブルの分割片の配置リストをオープンヘッダとして割符ファイルに追加することにより、割符ファイルに電子割符を生成することを特徴とする。

【０００８】
本発明の電子割符生成方法によれば、圧縮符号化と乱数を利用してたとえばビット単位の極く単純な手順を繰返すことにより生成される割符ファイルの中身は、元の情報とは全く異なる語順になっている。また、復元に用いる情報も切り刻まれ分割した状態で含まれているので、一部の割符ファイルが漏洩したときにも全部の変成情報を入手しなければ全く元の情報が復元できない。さらに、割符ファイルの中身を少しでも変化させれば電子割符に記載された手順を逆にたどっても意味のある電子情報に復元することができない。

【０００９】
したがって、本発明の方法により生成された電子割符を利用することにより、通信路中で電子割符を窃取しても利用することができない。また当事者が自己の保管する電子割符を改 しても相手の電子割符を統合したときに情報が復元できなくなるので、当事者間で電子割符を交換して復元することにより真正な情報を確認することができる。このように、本発明の電子割符生成方法をもちいることにより、高い信頼性を有する情報伝送が可能になる。

【００１０】
また、エレメント割振りテーブルをM個に分割するときに、予め決めた大きさのビット長を単位として分割し、乱数に基づいてM個の割符に分配するようにしてもよい。分割する語長はバイト単位であってもよい。
エレメント割り振りテーブルは、電子割符から元の情報を復元するために必須のものであるが、これを単に電子割符に添付するだけであれば、一部の電子割符を入手すれば、一部とはいえ元の情報が復元される可能性を残すことになる。
しかし、元の平文と同様に、テーブルを切り刻んで乱数を用いて順不同に電子割符に分配しておけば、全ての電子割符を収集しなければ情報の復元方法が解明できず、安全度が著しく向上する。

【００１１】
さらに、平文の先頭部分に適当な長さの乱数データを付加した後に圧縮符号化して符号語を作成するようにしてもよい。あるいは、平文を圧縮符号化した後に適当な長さの乱数データを付与して符号語としてもよい。
情報を摂取して利用しようとする敵は、様々な手法を駆使して元の情報を復元しようとするが、情報の先頭部に乱数が付与されていると、頭の方から解読しようとしても真の情報を見い出すことが困難になり先頭部からの攻撃を撃退することができる。また特に、乱数の長さを毎回変化させるようにする場合は、繰り返し攻撃される場合にも解読の糸口を与えず、解読の困難はさらに大きくなる。
なお、特に平文の語長が短い場合に、対象とする語長を長くして解読を困難にする効果が大きい。乱数データは２０から４０ビット程度でも十分安全である。

【００１２】
また、作成した符号語の先頭部分を適当な長さの乱数データでマスクして符号語を変成し、クローズヘッダにその乱数データを記録することが好ましい。
乱数データを発生して圧縮後データの先頭部に作用させてマスクする。たとえば両者の排

他的論理和（ＸＯＲ）をとることによりマスクすることができる。
このようなマスクは、元のデータに　味な部分を付加するので、さらに攻撃に対して耐性を強化することができる。乱数データが元の符号語と同じ長さであれば、理想的な強度を持つことになる。
【００１３】
なお、圧縮した符号語を分割したときのエレメントを１ビット長にする場合は、１ビットより短くすることができないため、振り分け分配の自由度が最も大きく、最も強固な防護能力がある。
また、エレメントをＭ個に割り振るときに使用する乱数はＮビットごとに同じ配列を繰返した疑似乱数であってもよい。
乱数の発生が簡単であり、電子割符を生成させるたびに乱数パターンが異なるので、ある桁ごとに繰返す疑似乱数であっても、十分安全性を確保することができる。また、熱雑音や宇宙線、あるいは無信号時にブラウン管に表示されるノイズなど、自然界に発生する雑音を利用することができる。なお、自然界のノイズには周期性がある場合もあるので、乱数性を確認した上で利用することが好ましい。
【００１４】
さらに、電子割符を３以上の配布先に配布するときに、配布先ごとに分配ファイルを準備して、Ｍ個の割符ファイルのそれぞれを異なる２個の分配ファイルに重複して含ませるように組合わせて、配布先に電子情報を分割配布するようにしてもよい。
この方法では、同じ電子割符が２個存在するので、配布された当事者の１が欠けたときにも、他の当事者から電子割符を集めることにより、欠落した電子割符を補充して元の割符を復元することができる。したがって、たとえば、相手方が契約の存在を否認する場合などでも、認証局から電子割符を提供させて復元することにより、元の契約内容を正しく復元して証明することができる。
なお、複数の記憶装置に分配しておいて後に収集して復元する場合にも、同様に方法を用いて元の契約内容を正しく復元して証明することができる。
【００１５】
また、本発明のコンピュータプログラムは、符号語ファイルと予め決めたＫ個のエレメントファイルと予め決めたＭ個の割符ファイルと平文を格納した平文ファイルとエレメント割振りテーブルをコンピュータメモリ上に準備する手順と、平文ファイルから平文を読み出して圧縮符号化し冗長なビットパターンを消した符号語を生成して符号語ファイルに格納する手順と、符号語を読み出してＫ個のエレメントに分割しエレメントごとにＫ個のエレメントファイルに格納する手順と、各エレメントを読み出しては乱数に基づいてＭ個の割符ファイルのいずれかに振り分けて格納する手順と、その振り分け方法をエレメント割振りテーブルに記録する手順と、そのエレメント割振りテーブルを読み出してＭ個に分割し割符ファイルにクローズヘッダとして追加する手順と、割符ファイルごとのクローズヘッダに分配された エレメント割振りテーブルの分割片の配置リストをオープンヘッダとして割符ファイルに追加する手順をコンピュータに実行させて、割符ファイルに電子割符を生成するようにすることを特徴とする。
【００１６】
このプログラムによりコンピュータは平文の提供を受けて簡単に電子割符を生成し提供することができる。
また、このプログラムを記録したコンピュータ読み取り可能な記憶媒体をコンピュータにセットしてプログラムを搭載させることにより、簡単にコンピュータを平文から電子割符を生成して配布するための装置とすることができる。
なお、このプログラムは通信回路網を介してダウンロードすることもできる。
【００１７】
【発明の実施の形態】
以下、本発明の電子割符生成方法およびプログラムを実施例に基づき図面を参照して詳細に説明する。

図１は、本発明の１実施例における電子割符生成方法の手順を示す流れ図、図２はその手順をデータ形態の変化に基づいて説明する概念説明図である。
本実施例の電子割符生成方法は、手順の概要を示す図１と、割符数を２、エレメント数を４として割符を作成する場合に生成されるデータの形態を説明した図２に示すように、まず、元データの平文Ｓ（ＡＢＣＤＥＦＧ・・）に圧縮処理などを施し、冗長なビットパターンを削除して符号語を生成する（Ｓ１）。

【００１８】

その後、符号語データを複数のエレメントＡＢ，ＣＤ，ＥＦ，Ｇ，・・に分割し（Ｓ２）、分割したエレメントを各割符▲１▼▲２▼に割り振り、割符中の格納位置を乱数に基づいて決定する（Ｓ３）。
その割り振り結果をメモリ上のエレメント割振りテーブルＴＡに記録し、さらに分割数や元データの長さなど情報の性格を知るために必要になる基本情報を加えてメモリ内のクローズヘッダファイルＣＨＦに記録する（Ｓ４）。

【００１９】

このクローズヘッダファイルＣＨＦの内容をバイト単位に切って、乱数に基づいて各割符▲１▼▲２▼に割り振り、クローズヘッダＣＨとして添付し（Ｓ５）、その割振り結果をメモリ上の復元情報振分けテーブルＴＢに記録する（Ｓ６）。割符番号など割符に関する基本情報と復元情報振分けテーブルＴＢをオープンヘッダＯＨとして割符に追加する（Ｓ７）。作成された割符データについて電子署名を作成し暗号化して電子割符を完成する（Ｓ８）。
完成した電子割符は、平文のエレメントの一部とエレメント割り振りテーブルの一部と復元情報振分けテーブルと割符の基本情報を含有している。
電子割符▲１▼▲２▼は、メモリ上の割符ファイルＦＴ１，ＦＴ２に格納されているので、これをフレキシブルディスクなどの記録媒体に出力したり、通信路を介して配布したりすることができる。

【００２０】

さらに、本実施例の電子割符生成方法をコンピュータに実行させるためのソフトウエアについて、図面を用いて説明する。
図３は本実施例のプログラムのメインフローを示す流れ図、図４はその元データ加工処理工程を説明する流れ図、図５はそのエレメント分割処理工程を説明する流れ図、図６はその割符データ編集処理工程を説明する流れ図、図７はその割符データ出力処理工程を説明する流れ図、図８は電子割符のレイアウト例を示す表である。

【００２１】

電子割符は、元データの加工処理プログラムＰ１と、エレメント分割処理プログラムＰ２と、割符データ編集処理プログラムＰ３と、割符データ出力処理プログラムＰ４を順に実行することにより生成され、所定の対象に対して発行される。
元データ加工処理プログラムＰ１は、元データの各ブロック単位で先頭部に乱数データを付与し、その後ブロック同士を連結する工程（Ｓ１１）と、得られたデータに圧縮処理を施す工程（Ｓ１２）と、圧縮したデータの先頭に圧縮処理に関する情報を圧縮ヘッダとして付加し（Ｓ１３）、適当な長さの乱数データを発生して圧縮データの先頭部とＸＯＲをとってマスクする工程（Ｓ１４）を備えて、平文から分割前の符号語を生成すると共に、使用した乱数データをメモリ内のファイルに格納しておく（Ｓ１５）。

【００２２】

このプログラムを実行すると、平文に乱数データを付与するため解号分析が困難になるが、特にデータ長の短い平文を対象とするときにもデータ長が増加して暗号解析を困難にする。侵害を意図する者が繰返し解析することにより解読できるようにさせないため、乱数長を処理ごとに変化させることが好ましい。なお、乱数データは２０～４０ビット程度であれば、情報の伝達は十分に安全である。また、圧縮後のデータの先頭部から適当な語長分をマスクすることにより、先頭部からの攻撃に高い耐性を備えることになる。マスクするために使用した乱数データはクローズヘッダに記録して後の割符復元処理に用いる。こ

の乱数の桁数は２０～４０ビットあれば十分であるが、圧縮データ全体のビット数と同じ長さにすれば理想的である。

【００２３】

エレメント分割処理プログラムＰ２は、符号語をエレメントに分割し電子割符に割付けると共に、割付情報を盛り込んだクローズヘッダを生成する手順を指示するものである。このプログラムは、図５に示すように、初めに、引数として入力したエレメント数で元データのビット数を割って得たビット数ごとに元データからエレメントを切り出す（Ｓ２１）。また、メモリ内のクローズヘッダファイルにソフトウエア認証キー、割符作成日時、元データのサイズ、割符数、エレメント数、エレメントの大きさなど分割処理に関する情報を記録する（Ｓ２２）。

【００２４】

さらに、引数の作成日時と元データのハッシュ値を利用して、乱数発生装置を割符数と同じ範囲の乱数列を初期化させ（Ｓ２３）、エレメントごとに乱数を用いて振分け先の割符を決定し、振分け位置情報をメモリ内のファイルに記録する（Ｓ２４）。元データのエレメントが尽きるまでエレメントの振分けを繰返す（Ｓ２５）。最終的な振分け情報はクローズヘッダファイルに記録される。

次に、クローズヘッダファイルを圧縮処理する（Ｓ２６）。

さらに、引数の作成日時とクローズヘッダのハッシュ値に基づいて乱数列を初期化し（Ｓ２７）、乱数を用いてクローズヘッドのバイトごとに振分け先の割符を決定し、振分け情報をメモリ内のファイルにテーブルとして記録する（Ｓ２８）。クローズヘッダはバイト数だけ繰返して割符への振分けがされる（Ｓ２９）。

【００２５】

割符データ編集処理プログラムＰ３は、図６に詳しく示したように、これまでに準備したデータから割符ファイルを作成するプログラムである。

まず、割符データを格納するための割符ファイルを作成する（Ｓ３１）。具体的には、１個の割符データを収納するメモリ領域を、指定された割符数分だけ確保することである（Ｓ３２）。

次に、電子割符ごとに、検索して電子割符を見つけだすために使用する検索用文字列、電子割符生成統合プログラムのバージョン番号、割符番号、クローズヘッダの長さなどを記載するオープンヘッダを作成しオープンヘッダファイルに格納する（Ｓ３３）。

【００２６】

さらに、クローズヘッダの振分け位置情報をバイトごとに確認し当該割符に配分されている場合はその番号をオープンヘッダファイルのリストに追加し（Ｓ３４）、これをクローズヘッダのバイト数だけ繰返す（Ｓ３５）。

次に、クローズヘッダファイルの内容を先の振分け情報に従って当該割符に配分されたバイトごとに割符ファイルのクローズヘッダの位置に記録する（Ｓ３６）。これをクローズヘッダファイルの内容のバイト数だけ繰返す（Ｓ３７）。

さらに、当該割符に割り付けられたエレメントを割符ファイルのデータ部に記録する（Ｓ３８）。これをエレメント数だけ繰返す（Ｓ３９）。

電子割符ごとに行われる上記工程を割符の数だけ繰返して（Ｓ４０）、全ての割符ファイルの作成を完了させる。

【００２７】

図７は、割符ファイルに記録される割符データのレイアウトを示す表である。

割符ファイルのオープンヘッダ部には、１．定長のテキスト文字列からなり割符か否かを識別するために用いる検索用文字列、２．使用する割符生成統合プログラムのバージョン番号、３．データサイズ、４．割符データの内容が変化していないかを確認するためのチェックサム値など、たとえばＭＤ５値、５．割符番号、６．クローズヘッダ部のサイズ、７．クローズヘッダ部の振分け位置情報の配列、など割符に関する基本的な情報が格納されている。

ただし、４．ＭＤ５値は後で実行される割符データ出力処理プログラムＰ４により生成さ

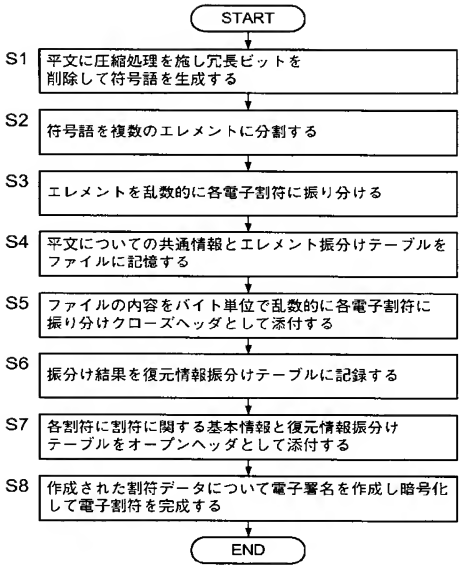
れるものである。
【００２８】
また、クローズヘッダ部には、８．ソフト認証キーに作用させて得たたとえばＭＤ５値、９．作成日時、１０．分割した元データのサイズ、１１．割符数、１２．エレメント数、１３．エレメントサイズ、１４．エレメントの振分け位置情報の配列、など元データに関する基本的な情報が格納されている。
なお、クローズヘッダ部の情報は、既に分割して配分された状態になっているので、１個の割符データの中にこれら情報が画然と配置されているわけではないことはいうまでもない。
さらにデータ部には、１５．エレメント自体が配列された状態で収納されている。
【００２９】
割符データ出力処理プログラムＰ４は、図８に示すように、割符ファイルから電子割符を生成し、ファイルまたはメモリに出力して利用できる形態にするためのプログラムである。
初めに、割符データのファイルの１．定長のテキスト文字列からなり割符か否かを識別するために用いる検索用文字列、２．使用する割符生成統合プログラムのバージョン番号、３．データサイズを残して、５．割符番号を含めて後ろの部分についてチェックサムアルゴリズムたとえばＭＤ５を作用させてチェックサムを求め、割符ファイルのオープンヘッダ部に４．チェックサム値として追加する（Ｓ５１）。
次に、引数として入力される本人確認ができる共通鍵キーを用いて割符ファイルの４．ＭＤ５値以降の部分を暗号化して（Ｓ５２）、ファイルまたはメモリに出力する（Ｓ５３）。
これを割符の数だけ繰返すことにより、電子割符を生成する（Ｓ５４）。
【００３０】
電子割符から元データを復元するときは、電子割符に検索用文字列がそのまま見ることができる状態で添付されているので、色々なデータが混在しているところから必要な電子割符を抽出して収集することができる。全ての電子割符が集ると、割符生成時に指定した本人確認キー、割符生成時に指定したときはソフト認証キーや作成日時、などを用いて、割符生成と逆の手順を踏むことにより元のデータを復元することができる。
【００３１】
本実施例の電子割符生成方法において、割符数、エレメント数、テーブルの分割数などは、電子割符生成上のパラメータであって、任意の数を指定することができる。また、乱数の発生や初期化には周知の方法を利用すればよく、適当な長さの乱数列を繰返して生成する疑似乱数を利用してもよい。
さらに、割符データには、使用の目的や安全性の要求などに対応して、列挙した事項の全てを含まなくてもよく、また別の事項を含んでもよいことはいうまでもない。
【００３２】
なお、本願発明において利用する電子割符法は、秘密情報を分割して安全に伝送あるいは記録するための符号化法で、秘密情報である平文Ｓをｎ個の割符Ｗｉに分割符号化し、ｎ個の割符が全部そろえば平文Ｓが復元できるが、ｎ－１個以下の割符からは平文Ｓの情報が漏れないようになっている。この方法は、（ｎ，ｎ）しきい値秘密分散法あるいは（ｎ，Ｌ，ｎ）しきい値ランプ型秘密分散法の特殊な場合と考えることができる。
一般的な秘密分散法は、ガロア体上の演算を用いて実現されることが多いが、電子割符法ではビット単位の平文の割り振りやマスク処理だけですむため高速な処理が可能であり、またかなり大きなファイルであっても分割することなく一度に符号化ができることなどに特長がある。
【００３３】
次に、本実施例の電子割符生成法で生成した電子割符の攻撃耐性について説明する。
まず、敵が１個を除きｎ－１個の割符を入手したような最も危険な場合を考える。電子割符では平文Ｓについて圧縮符号化しているため、ｎ－１個の割符からは平文Ｓのどの部分

も直ぐには求めることができない。
そこで、残りの未入手の割符について全数探索して平文を復元するためには、符号語の語長をＮとし、符号語の割符への割り振り操作の回数をＭとし、ｍを$2^{M}/(M+1)^{n}$以上の数とすると、秘密分散法においてｎ－１個の割符が知られたときの平文の　昧さが（Ｎ＋ｍ）／ｎと評価されることから、２の（Ｎ＋ｍ）／ｎ乗回の探索をしなければならないことになる。この数は語長Ｎの増加と共に指数関数的に増加するので、ある程度大きな平文Ｓに対しては攻撃が全く不可能ということができる。
【００３４】
また、敵がｎ－１個の割符から平文Ｓの一部でも復元することができるかを検討すると、平文Ｓについて逐次的な圧縮符号化がされた場合でもＳの最初のビットを推定するために２のｍ／ｎ乗通りの探索を行わなければならないのに、ビット長ｒ１の乱数でマスク処理を行っているので、２の（ｍ＋ｒ１）／ｎ乗通りの探索が必要となる。
さらに、全数探索攻撃では、復合されたデータが意味のあるものかどうかを判断しなければならないが、平文Ｓの前にビット長ｒ２の乱数が付加されているので、ｒ２ビット取り込んだ後でなければ判断ができないから、結局、２の（ｍ＋ｒ１＋ｒ２）／ｎ乗通りの探索を行わなければならないことになる。
たとえばｎ＝８のとき、Ｍ＝ｒ１＝ｒ２＝５１２ビット（６４バイト）であれば、2^{18}すなわち10^{55}、Ｍ＝ｒ１＝ｒ２＝１０２４ビットであれば、2^{374}すなわち10^{112}という莫大な回数の探索が要求される。したがって、全数探索により平文Ｓの最初の数ビットの値を知ることは実際上不可能といえ、十分な安全性があることが分る。
【００３５】
さらに、大きなブロック単位で符号化する圧縮符号化法を用いる場合は、そのブロック長単位で解読する必要があり、未入手の割符中の情報を含めて全数探索しなければならなくなり、必要な探索数がさらに飛躍的に増加する。
敵が、複数の割符を入手して、それらに共通する手順を探ろうとする攻撃が考えられる。しかし、電子割符では各処理において使い捨ての乱数列を使用するので、同じ平文でも生成の都度異なる割符となるため、このような攻撃は不可能である。
割符の長さから平文の長さを測ろうとしても、不定長の乱数データをダミーとして付加した上圧縮処理がなされるため、割符の長さと平文の長さに直接の関係がない。
このように、本実施例により得られる電子割符は極めて高度な安全性を備えている。
【００３６】
なお、上記説明した本実施例の電子割符は、一つでも紛失したり改　されたりすると原本を復元できなくなる。
したがって、各電子割符をｎカ所で分散して保管する場合に、その保管責任は重大であり、また一カ所でも　損を受ければ正しい復元ができないので、危険性はｎ倍になることになる。さらに、当事者間に利害関係を有する原本では一方が電子割符の所有を否認することすら考えられる。
そこで、全部でなくｎ－１個の電子割符を収集すれば復元できるように改変すれば、このような不都合を防止することができる。
【００３７】
図９は、Ｘ，Ｙ，Ｚの３カ所に電子割符を配布し、必要なときに３カ所から電子割符を集めて元の情報を復元するようにした場合において、１個の電子割符が　損されても残りの２個から正しく復元することができるようにしたものを示す。
上記説明した電子割符生成法に従って生成した割符ファイルＡ，Ｂ，Ｃを複写して２セット作成し、たとえばＸにＡ＋Ｃ、ＹにＢ＋Ａ、ＺにＣ＋Ｂというように、互いに重複しないように組合わせて分配する。
【００３８】
したがって、いずれの１カ所の情報が欠落しても残りの２カ所から集めた情報に全部の割符ファイルが含まれるので、元の情報を正しく復元することができる。
なお、内部グループ化したハッシュ値を格納しておけば、情報自体を復元しなくても残り

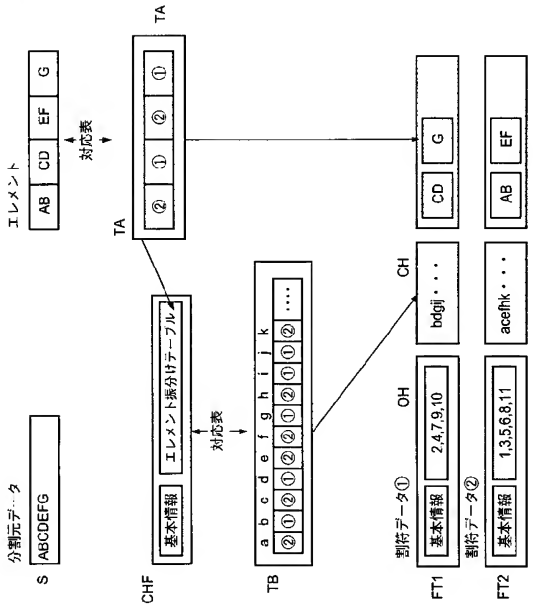
の１個の原本性を検証することができる。
図９の方法を用いれば、たとえば、ＸとＹで交した契約を３個の電子割符に分割して、１個を認証サービス会社Ｚに保管させておき、争いが生じたときはＺから取寄せた電子割符を使って真正の契約書を復元して確認することができる。
【００３９】
また、この方法は１個の保管データが　損されても残りのデータから正しい情報が復元できることから、非常時のバックアップ機能として活用することもできる。
図１０は、認証サービス会社が保管するデータをバックアップする場合を説明する図面である。
【００４０】
顧客Ｘと販売会社Ｙと認証サービス会社Ｚが３個の電子割符をそれぞれ保管する。認証サービス会社Ｚは、自己の責任において保管する電子割符Ｚをさらに電子割符化して割符ファイル（Ａ＋Ｂ＋Ｃ）を作成し、これを図９のように分配して３個の新しい電子割符Ｚ１（Ａ＋Ｃ），Ｚ２（Ｂ＋Ａ），Ｚ３（Ｃ＋Ｂ）とし、それぞれ別のサーバに分けて保管する。
顧客Ｘあるいは販売会社Ｙから照会があったときには、認証サービス会社Ｚはサーバから電子割符を収集して元の情報を復元して提供する。このとき、１個の電子割符に損傷があっても残りの電子割符から正しい情報を復元することができる。
【００４１】
なお、図１１に示すように、平文から作成する割符ファイルを５個（Ａ，Ｂ，Ｃ，Ｄ，Ｅ）にしたときは、各所にたとえば（Ａ＋Ｅ），（Ｂ＋Ａ），（Ｃ＋Ｂ），（Ｄ＋Ｃ），（Ｅ＋Ｄ）というように分配すれば、いずれか４個の電子割符があれば全ての割符ファイル（Ａ，Ｂ，Ｃ，Ｄ，Ｅ）が集るので正しく元の情報を復元することができる。
すなわち、ｎ個の電子割符を作成しｎカ所に分配したときには（ｎ－１）個の電子割符を集めれば元の情報を復元することができるわけである。
【００４２】
【発明の効果】
以上詳細に説明した通り、本発明の電子割符生成方法および電子割符生成プログラムを使用すると、秘密分散法に準じた高度な安全性を有する電子割符を簡単に作成することができ、通信・保管の安全や契約の安定を確保することができる。
【図面の簡単な説明】
【図１】本発明の１実施例における電子割符生成方法の手順を示す流れ図である。
【図２】本実施例の手順をデータ形態の変化に基づいて説明する概念説明図である。
【図３】本実施例に使用するプログラムのメインフローを示す流れ図である。
【図４】本実施例のプログラムにおける元データ加工処理を説明する流れ図である。
【図５】本実施例のプログラムにおけるエレメント分割処理を説明する流れ図である。
【図６】本実施例のプログラムにおける割符データ編集処理を説明する流れ図である。
【図７】本実施例のプログラムにおける割符データ出力処理を説明する流れ図である。
【図８】本実施例の生成方法により得られる電子割符のレイアウト例を示す表である。
【図９】本実施例の電子割符生成方法の別の利用態様を説明する図面である。
【図１０】図９の利用態様の適用例を説明するブロック図である。
【図１１】図９の利用態様の別の形態を説明する図面である。
【図１２】本発明に使用する電子割符を説明する概念図である。
【符号の説明】
ＣＨ　クローズヘッダ
ＣＨＦ　クローズヘッダファイル
ＦＴ１，ＦＴ２　割符ファイル
ＯＨ　オープンヘッダ
Ｓ　平文
ＴＡ　エレメント割振りテーブル

ＴＢ　復元情報振分けテーブル

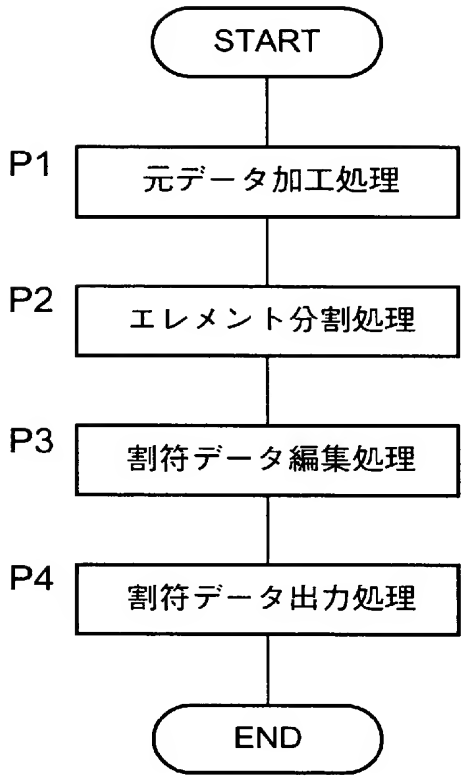
【図１】



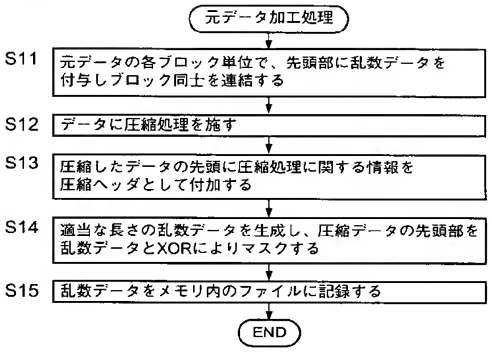
【図２】



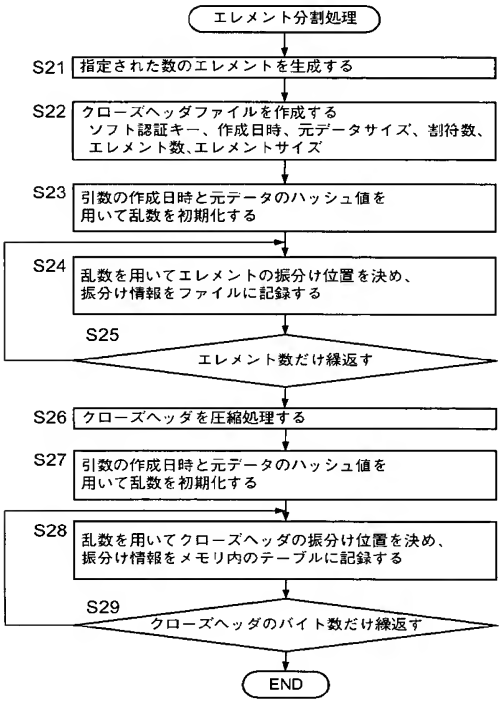
【図３】


START
P1
元データ加工処理
P2
エレメント分割処理
P3
割符データ編集処理
P4
割符データ出力処理
END

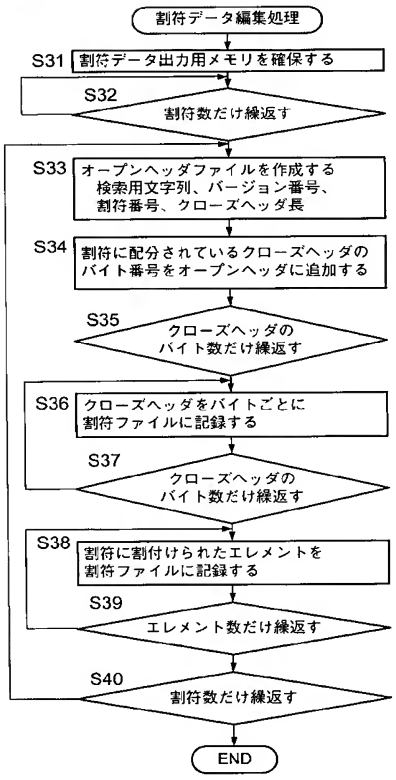
【図４】


元データ加工処理
S11
元データの各ブロック単位で、先頭部に乱数データを付与しブロック同士を連結する
S12
データに圧縮処理を施す
S13
圧縮したデータの先頭に圧縮処理に関する情報を圧縮ヘッダとして付加する
S14
適当な長さの乱数データを生成し、圧縮データの先頭部を乱数データとXORによりマスクする
S15
乱数データをメモリ内のファイルに記録する
END

【図５】


エレメント分割処理
S21
指定された数のエレメントを生成する
S22
クローズヘッダファイルを作成する
ソフト認証キー、作成日時、元データサイズ、割符数、エレメント数、エレメントサイズ
S23
引数の作成日時と元データのハッシュ値を用いて乱数を初期化する
S24
乱数を用いてエレメントの振分け位置を決め、振分け情報をファイルに記録する
S25
エレメント数だけ繰返す
S26
クローズヘッダを圧縮処理する
S27
引数の作成日時と元データのハッシュ値を用いて乱数を初期化する
S28
乱数を用いてクローズヘッダの振分け位置を決め、振分け情報をメモリ内のテーブルに記録する
S29
クローズヘッダのバイト数だけ繰返す
END

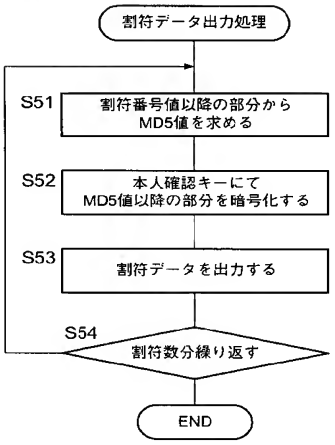
【図６】


割符データ編集処理
S31
割符データ出力用メモリを確保する
S32
割符数だけ繰返す
S33
オープンヘッダファイルを作成する
検索用文字列、バージョン番号、割符番号、クローズヘッダ長
S34
割符に配分されているクローズヘッダのバイト番号をオープンヘッダに追加する
S35
クローズヘッダのバイト数だけ繰返す
S36
クローズヘッダをバイトごとに割符ファイルに記録する
S37
クローズヘッダのバイト数だけ繰返す
S38
割符に割付けられたエレメントを割符ファイルに記録する
S39
エレメント数だけ繰返す
S40
割符数だけ繰返す
END

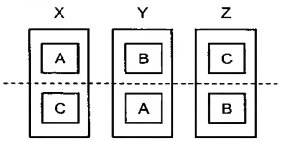
【図７】

番号	格納内容
オープンヘッダ部	
1	検索用文字例(割符データを識別するためのテキスト文字列)
2	バージョン番号
3	データサイズ
以降は、暗号化(ユーザ指定の共通鍵キー)されて出力される	
4	MD5値
5	割符番号
6	クローズヘッダ部のサイズ(各割符共通の内容がセットされる)
7	クローズヘッダ部の振分位置情報の配列(各割符共通の内容がセットされる)
クローズヘッダ部	
8	ソフト認証キーのMD5値
9	作成日時(生成時に指定された値で乱数を初期化する、統合時に指定されたら照合する)
10	分割元データサイズ
11	割符数
12	エレメント数
13	エレメントサイズ
14	エレメントの振分位置情報の配列 (生成時に本割符データに振り分けられた元データのエレメント内容を格納)
データ部	
15	エレメントの配列 (生成時に本割符データに振り分けられた元データのエレメント内容を格納)

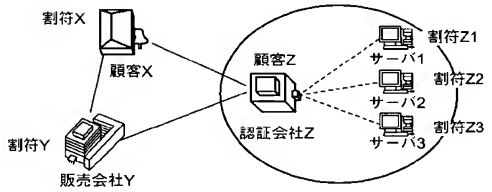
【図８】



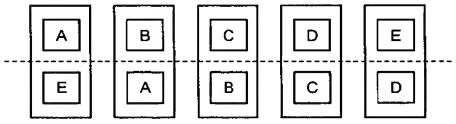
【図９】



【図１０】



【図１１】



【図１２】